This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/671,875	09/28/2000	Geoffrey Owen Blandy	AUS9-2000-0587-US1	7126

35525 7590 08/25/2004

IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
TSAI, HENRY	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 08/25/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/671,875	BLANDY, GEOFFREY OWEN	
	Examiner	Art Unit	
	Henry W.H. Tsai	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *6/1/04 and 3/31/04* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) __ is/are allowed.

6)☒ Claim(s) *1-5,7-18,20-31 and 33-39* is/are rejected.

7)☒ Claim(s) *6, 19, and 32* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on __ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *6/1/04 / 3/31/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Claim Objections

1. Claims 5, 7, 8, 11, 20, 21, 24, 33, 34, and 37 are objected to because of the following informalities:

In claim 5, line 3, "is" should read -are- since the steps is plural.

In claim 7, the definition of "a most common combination" was not defined previously since how to set up a "common" level is unclear. Similar problems exist in claims 20, and 33.

In claim 8, the definition of "a most restrictive" and "less restrictive" were not defined previously since how to set up a restrictive level is unclear. Similar problems exist in claims 21, and 34.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless -
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-5, 7-18, 20-31, and 33-39 are rejected under 35 U.S.C. 102(b) as being anticipated by Hull et al. (USP 5,922,065), herein referred to as Hull et al.'065, and Berenbaum et al. (USP 6,658,551), herein referred to as Berenbaum et al.'551.

Referring to claims 1, 14, and 27, Hull et al.'065 discloses as claimed, a method for creating instruction bundles (comprising slot0, slot1 and slot2, see Fig. 4), comprising: receiving an instruction group (see Col. 4, lines 1-2) having one or more instructions (comprising the instruction types in the table 20 as shown in Fig. 2); automatically determining a number of each possible type of instruction (instruction type such as A, I-unit, M-unit, and F-unit see Figs. 2) in the one or more instructions of the instruction group (see Fig. 4, the instructions are grouped into different groups separated by the double lines 42 and 43; and the stop bits (S-bit see Fig. 3),

see also Col. 4, lines 43-45, and lines 61-67, each group comprises different execution unit types, such as I-unit, M-unit, and F-unit see Figs. 2 and 4 based on the architectural limitations thereof); and dynamically creating one or more instruction bundles (comprising slot0, slot1 and slot2, see Fig. 4) based on the number of each possible type of instruction in the one or more instructions of the instruction group (as set forth above, see Fig. 4, the instructions are grouped into different groups separated by the double lines 42 and 43; and the stop bits (S-bit see Fig. 3), see also Col. 4, lines 43-45, and lines 61-67, each group comprises different execution unit types, such as I-unit, M-unit, and F-unit see Figs. 2 and 4 based on the architectural limitations thereof). Note Hull et al.'065 teaches the automatically determining and the dynamically creating steps since it inherently comprises an optimizing compiler to execute the steps in order to generate the bundles and groups as shown in Fig. 4. As mentioned at Col. 2, lines 12-15 in Berenbaum et al.'551's reference that a VLIW computer relies primarily on an optimizing compiler to generate binaries capable of exploiting the VLIW processor's multiple issue capabilities.

As to claims 2, 15, and 28, Hull et al. also discloses: receiving an instruction group having one or more instructions

includes receiving a stream of intermediate instructions (comprising the instruction types in the table 20 as shown in Fig. 2) organized into instruction groups.

As to claims 3, 16, and 29, Hull et al. also discloses: gathering information about an architecture for use in creating instruction bundles (comprising slot0, slot1 and slot2, see Fig. 4. Note each group comprises different execution unit types, such as I-unit, M-unit, and F-unit see Figs. 2 and 4 based on the architectural limitations thereof).

As to claims 4, 17, and 30, Hull et al. also discloses: the information includes at least one of a number of each type of execution unit available in the architecture and a number of bundles (as set forth, each bundle comprising slot0, slot1 and slot2, see Fig. 4. Note each group comprises different execution unit types, such as I-unit, M-unit, and F-unit see Figs. 2 and 4 based on the architectural limitations thereof) that can be dispatched concurrently by the architecture (see Col. 4, lines 1-2, regarding "a instruction group is a set of statically contiguous instructions that may be executed concurrently").

As to claims 5, 18, and 31, Hull et al. also discloses: the steps of determining a number of each possible type of instruction and creating one or more instruction bundles is performed for each instruction group in the stream of

intermediate instructions (comprising the instruction types in the table 20 as shown in Fig. 2).

As to claims 7, 20, and 33, Hull et al. also discloses, creating one or more instruction bundles is performed based on a most common instruction combination first (note each bundle comprises different most common instruction combination using execution unit types, such as I-unit, and M-unit, see Figs. 2 and 4 based on the architectural limitations thereof).

As to claims 8, 21, and 34, Hull et al. also discloses, as best understood: creating one or more instruction bundles is performed based on a most restrictive instruction type placement (such as the bundle with template 0 having M-unit since memory access should be processed first to fetch operands) and proceeds to less restrictive instruction type placement (such as the bundle with template 0 having I-unit since integer operation needs operands to be fetched first) second.

As to claims 9, 22, and 35, Hull et al. also discloses: creating one or more instruction bundles is performed based on a most restrictive instruction type placement (such as the bundle with template 0 having M-unit since memory access should be processed first to fetch operands) and proceeds to less restrictive instruction type placement (such as the bundle with

template 0 having I-unit since integer operation needs operands to be fetched first).

As to claims 10, 23, and 36, Hull et al. also discloses: determining a number of each possible type of instruction in the one or more instructions of the instruction group includes incrementing instruction counters (note the instruction counters are considered as inherent elements in Hull et al.'065 in order to measure or control the number of instruction generated in each bundle and group by an optimizing compiler (see also Col. 2, lines 12-15 in Berenbaum et al.'551's reference); further see Col. 5, lines 15-18, regarding "bundles are ordered from lower to highest memory address. Instructions in bundles with lower memory address are considered to precede instructions in bundles with higher memory address"; see also Col. 5, lines 20-23, regarding the instruction group being ordered (best broadly and reasonably interpreted as "counted") from instruction slot 0 to instruction slot 2 as shown in Fig. 3) based on the number of each possible type of instruction in the one or more instructions, and wherein creating one or more instruction bundles includes decrementing (note as mentioned above, incrementing and decrementing the instruction counters are the inherent steps in order to measure or control the number of instruction generated in each bundle and group by an optimizing

compiler) the instruction counters as instructions are added to instruction bundles.

As to claims 11, 24, and 37, Hull et al. also discloses: the most common instruction combination is where all instructions in the instruction group are of a memory instruction type, integer instruction type or integer arithmetic logic unit type (note each bundle comprises different most common instruction combination using execution unit types, such as I-unit, and M-unit, see Figs. 2 and 4 based on the architectural limitations thereof).

As to claims 12, 25, and 38, Hull et al. also discloses: creating one or more instruction bundles includes ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription (note each bundle comprises different most common instruction combination using execution unit types, such as I-unit, and M-unit, see Figs. 2 and 4 based on the architectural limitations thereof to avoid hardware oversubscription).

As to claims 13, 26, and 39, Hull et al. also discloses: ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription includes forming partial instruction bundles (see Fig. 4, the instruction bundles with templates 1 and 5 having double lines 42 and 43 used as a

separator for different groups, the instruction bundles with templates 1 and 5 are best reasonably and broadly interpreted as a partial instruction bundle to avoid hardware oversubscription).

Allowable Subject Matter

4. Claims 6, 19, and 32 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

5. Applicant's arguments mailed 3/31/04 have been considered but are moot in view of the new ground(s) of rejection.

Applicant argues that "there is no teaching anywhere in the **Hull reference as to determining a number of each possible type of instruction in the one or more instructions of the instruction group" (page 14, lines 20-22).** Examiner disagrees with Applicant. As set forth in the art rejection above, Hull et al.'065 discloses the step of: automatically determining a

number of each possible type of instruction (instruction type such as A, I-unit, M-unit, and F-unit see Figs. 2) in the one or more instructions of the instruction group (see Fig. 4, the instructions are grouped into different groups separated by the double lines 42 and 43; and the stop bits (S-bit see Fig. 3), see also Col. 4, lines 43-45, and lines 61-67, each group comprises different execution unit types, such as I-unit, M-unit, and F-unit see Figs. 2 and 4 based on the architectural limitations thereof). Note Hull et al.'065 teaches the automatically determining and the dynamically creating steps since it inherently comprises a VLIW compiler to execute the steps in order to generate the bundles and groups as shown in Fig. 4. As mentioned at Col. 2, lines 12-15 in Berenbaum et al.'551's reference that a VLIW computer relies primarily on an optimizing compiler to generate binaries capable of exploiting the VLIW processor's multiple issue capabilities.

Applicant also argues that "nowhere in this section, or any **other section of Hull, are instruction counters incremented based on the number of each possible type of instruction in the one or more instructions or are instruction counters,** decremented as instructions are added to instruction bundles" **(page 17, last two lines, and page 18, lines 1-2); and** "moreover, nowhere **is a counter even mentioned in this or any**

other section of Hull, let alone Incrementing/decrementing counters in the manner recited" (page 18, lines 4-5). Examiner disagrees with Applicant. As set forth in the art rejection above, Hull et al.'065 discloses: determining a number of each possible type of instruction in the one or more instructions of the instruction group includes incrementing instruction counters (note the instruction counters are considered as inherent elements in Hull et al.'065 in order to measure or control the number of instruction generated in each bundle and group by an optimizing compiler (see also Col. 2, lines 12-15 in Berenbaum et al.'551's reference); further see Col. 5, lines 15-18, regarding "bundles are ordered from lower to highest memory address. Instructions in bundles with lower memory address are considered to precede instructions in bundles with higher memory address"; see also Col. 5, lines 20-23, regarding the instruction group being ordered (best broadly and reasonably interpreted as "counted") from instruction slot 0 to instruction slot 2 as shown in Fig. 3) based on the number of each possible type of instruction in the one or more instructions, and wherein creating one or more instruction bundles includes decrementing (note as mentioned above, incrementing and decrementing the instruction counters are the inherent steps in order to measure or control the number of instruction generated in each bundle

and group by an optimizing compiler) the instruction counters as instructions are added to instruction bundles.

Applicant further argue nowhere, in any section of Hull, is it taught: creating one or more instruction bundles is performed based on a most common instruction combination first, as in claims 7, 20 and 33; creating one or more instruction bundles is performed based on a most restrictive instruction type placement and proceeds to less restrictive instruction type placement second as in claims 8, 21 and 34; creating one or more instruction bundles is performed based on a most restrictive instruction type placement and proceeds to less restrictive instruction type placement, as in claims 9, 22 and 35; where the most common instruction combination is where all instructions in the instruction group are of a memory instruction type, integer instruction type or integer arithmetic logic unit type, as in claims 11, 24 and 37; where creating one or more instruction bundles ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription, as in claims 12, 25 and 38; or ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription includes forming partial instruction bundles, as in claims 13, 26 and 39 (page 18, lines 17-30). Examiner disagrees with Applicant. As set forth in the art rejection above, with

respect to claims 7, 20, and 33, Hull et al. also discloses, creating one or more instruction bundles is performed based on a most common instruction combination first (note each bundle comprises different most common instruction combination using execution unit types, such as I-unit, and M-unit, see Figs. 2 and 4 based on the architectural limitations thereof);

As to claims 8, 21, and 34, Hull et al. also discloses: creating one or more instruction bundles is performed based on a most restrictive instruction type placement (such as the bundle with template 0 having M-unit since memory access should be processed first to fetch operands) and proceeds to less restrictive instruction type placement (such as the bundle with template 0 having I-unit since integer operation needs operands to be fetched first) second.

As to claims 9, 22, and 35, Hull et al. also discloses: creating one or more instruction bundles is performed based on a most restrictive instruction type placement (such as the bundle with template 0 having M-unit since memory access should be processed first to fetch operands) and proceeds to less restrictive instruction type placement (such as the bundle with template 0 having I-unit since integer operation needs operands to be fetched first);

As to claims 10, 23, and 36, Hull et al. also discloses: determining a number of each possible type of instruction in the one or more instructions of the instruction group includes incrementing instruction counters (note the instruction counters are considered as inherent elements in Hull et al.'065 in order to measure or control the number of instruction generated in each bundle and group by an optimizing compiler (see also Col. 2, lines 12-15 in Berenbaum et al.'551's reference); further see Col. 5, lines 15-18, regarding "bundles are ordered from lower to highest memory address. Instructions in bundles with lower memory address are considered to precede instructions in bundles with higher memory address"; see also Col. 5, lines 20-23, regarding the instruction group being ordered (best broadly and reasonably interpreted as "counted") from instruction slot 0 to instruction slot 2 as shown in Fig. 3) based on the number of each possible type of instruction in the one or more instructions, and wherein creating one or more instruction bundles includes decrementing (note as mentioned above, incrementing and decrementing the instruction counters are the inherent steps in order to measure or control the number of instruction generated in each bundle and group by an optimizing compiler) the instruction counters as instructions are added to instruction bundles;

As to claims 11, 24, and 37, Hull et al. also discloses: the most common instruction combination is where all instructions in the instruction group are of a memory instruction type, integer instruction type or integer arithmetic logic unit type (note each bundle comprises different most common instruction combination using execution unit types, such as I-unit, and M-unit, see Figs. 2 and 4 based on the architectural limitations thereof);

As to claims 12, 25, and 38, Hull et al. also discloses: creating one or more instruction bundles includes ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription (note each bundle comprises different most common instruction combination using execution unit types, such as I-unit, and M-unit, see Figs. 2 and 4 based on the architectural limitations thereof to avoid hardware oversubscription); and

As to claims 13, 26, and 39, Hull et al. also discloses: ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription includes forming partial instruction bundles (see Fig. 4, the instruction bundles with templates 1 and 5 having double lines 42 and 43 used as a separator for different groups, the instruction bundles with templates 1 and 5 are best reasonably and broadly interpreted as

a partial instruction bundle to avoid hardware oversubscription).

In summary, as set forth in the art rejections above, Hull et al.'065 and Berenbaum et al.'551 teach the claimed invention.

Conclusion

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dr. Henry Tsai whose telephone number is (703) 308-7600. The examiner can normally be reached on Monday-Thursday from 8:00 AM to 5:00 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner supervisor, Eddie Chan, can be reached on (703) 305-9712. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the **TC 2100 receptionist whose telephone number is (703) 305-3900.**

8. In order to reduce pendency and avoid potential delays, Group 2100 is encouraging FAXing of responses to Office actions directly into: **the Group at fax number: 703-872-9306.**

This practice may be used for filing papers not requiring a fee. It may also be used for filing papers which require a fee by applicants who authorize charges to a PTO deposit account.

Please identify the examiner and art unit at the top of your cover sheet. Papers submitted via FAX into Group 2100 will be promptly forward to the examiner.

HENRY W. H. TSAI
PRIMARY EXAMINER
August 18, 2004